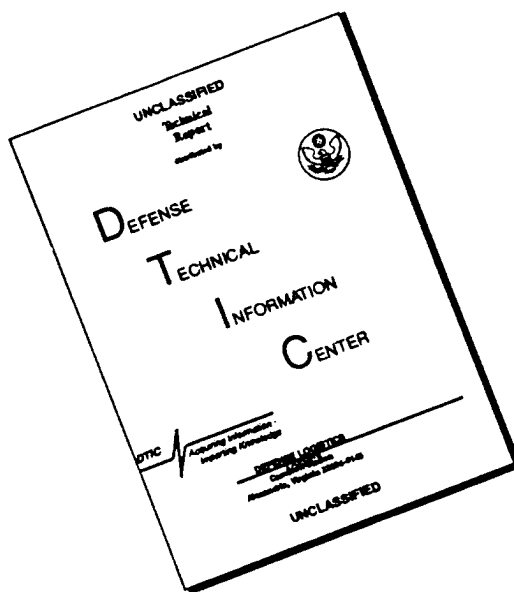


REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE July 24, 1996		3. REPORT TYPE AND DATES COVERED Final 3/01/93 - 02/28/96
4. TITLE AND SUBTITLE Semantic Theories and Automated Tools for Real-Time and Probabilistic Concurrent Systems			5. FUNDING NUMBERS F49620-93-1-0169	
6. AUTHOR(S) Professor Amy E. Zwarico				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) The Johns Hopkins University 3400 N. Charles Street Baltimore, MD 21218			8. PERFORMING ORGANIZATION REPORT NUMBER AFOSR TR 96-0419	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NM 110 Duncan Ave, Suite B115 Bolling AFB DC 20332			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES			19960813 168	
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release: Distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The results fall into two primary categories: real-time process algebras and type theories for object oriented programs. The process algebra results include a richer semantic framework for expressing real time processes, the development of an algebra for expressing both preemptive and nonpreemptive processes, axiomatizations of the algebra, implementations of these, axiomatizations, and equivalence for Pict, a concurrent object-oriented programming language based on the π -calculus. The other results include the development of constraint-based type systems (including inference algorithms) for object-oriented languages.				
14. SUBJECT TERMS			15. NUMBER OF PAGES	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED			18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	
19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED			20. LIMITATION OF ABSTRACT SAR	

DTIC QUALITY INSPECTED 4

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

JUL-25-1996 15:58

U COMP SCI

410 516 5134 P.003/004

Final Progress Report: Semantic Theories and Automated Tools for Real-Time and Probabilistic Concurrent Systems

1 Results

The results fall into two primary categories: real-time process algebras and type theories for object oriented programs. The process algebra results include a richer semantic framework for expressing real-time processes, the development of an algebra for expressing both pre-emptive and nonpre-emptive processes, axiomatizations of the algebras, implementations of these axiomatizations, and equivalences for Pict, a concurrent object-oriented programming language based on the π -calculus. The OO results include the development of constraint-based type systems (including inference algorithms) for object-oriented languages.

- We extended the class of nonpre-emptive processes first developed in [?]; revised the alternative characterization of nonpre-emptive processes; defined both real-time and nonpre-emptive algebras; and axiomatized the timed preorders. The extension of the class of nonpre-emptive processes allows us to represent more nondeterministic processes, thus increasing the expressivity of the formalism. The real-time algebra, in conjunction with the testing preorders, allows us to express general timing constraints such as upper and lower bounds. The nonpre-emptive algebra is useful because it can only be used to express nonpre-emptive processes. Furthermore, the nonpre-emptive preorders are precongruences for all operators of the algebra. We have used results from [?, ?] to develop the axiomatizations for both the general real-time algebra and the nonpre-emptive algebra.
- We currently have a Standard ML implementation of BDD's for the real-time algebras developed. We are using Emerson's quantitative μ -calculus [?], to specify formulas which are to be checked in the model.
- We are developing equivalences for Pict, a concurrent object-oriented programming language based on the π -calculus. Pict consists of a core language in which higher-level language can be defined. Proving equivalence of Pict programs consists in first developing an equivalence for core Pict, developing translation rules that map high-level constructs into low level ones, and proving that the translation rules preserve program equivalence. Thus if two high-level programs are equivalent then so are their core-level translations.
- The development of rich type systems for object-oriented programming languages. Our main area of interest is constraint-based type inference for object-oriented languages.

Type inference, the idea of automatically inferring type information from untyped programs, is originally due to Hindley and Milner. The type inference for object-oriented languages is more difficult than for traditional sequential programs because even simple object-oriented programs are more "advanced" type-theoretically. Object types can be self-referential so some form of recursive type is also needed. This can be captured using a form of operator polymorphism or F-bounded polymorphism.

We have defined a polymorphic, constraint-based type inference algorithm for an object-oriented language, I-Loop. This language incorporates standard notions of class, multiple inheritance, object creation, message send, mutable instance variables, and hiding of instance variables. Thus, there is enough of a core that we are reasonably confident the ideas will scale

JUL-25-1996 15:59

U COMP SCI

410 516 6134 P.004/004

up to an implemented language. We infer a generalized form of type, recursively constrained (rc) types of the form $\tau \ C$, with "reading" "where." C is a set of type constraints of the form $\tau <: \tau'$ possibly containing free type variables. These constraints may be recursive in that a variable t could occur free in both τ and τ' . This form of type is not standard, and generalizes existing notions. The rc types have the advantage of being very expressive, but the disadvantage of being more difficult to read. The recursive constraints generalize recursive types and the rd-polymorphism generalizes F-bounded polymorphism. The constrained type inference approach to objects allows the inference of very detailed rc types. Far more precise than any type a programmer is likely to come up with. And, this will in turn give the programmer a degree of flexibility well beyond what is provided by current typed object-oriented languages.

2 Papers

- Elizabeth Leonard and Amy Zwarico, "an Algebraic Framework for Developing and Maintaining Real-Time Systems," AMAST '95, Lecture Notes in Computer Science, vol. 936, Springer Verlag.
- Kim B. Bruce, Luca Cardelli, Giuseppe Castagna, Scott F. Smith, Jonathan Eifrig, Valery Trifonov, Tiejun Wang, Gary T. Leavens, and Benjamin Pierce. On Binary Methods. Submitted for publication. Department of Computer Science, Iowa State University, TR95-08/A1, May 1995.
- J. Eifrig, S. Smith, V. Trifonov, "Sound Polymorphic Type Inference for Objects," OOPSLA 1995.
- J. Eifrig, S. Smith, V. Trifonov, "Type Inference for Recursively Constrained Types and its Application to OOP", Mathematical Foundations of Programming Semantics 1995 (Elsevier Electronic Notes in Theoretical Computer Science, volume 1).
- J. Eifrig, S. Smith, V. Trifonov, A. Zwarico, "Application of OOP Type Theory: State, Decidability, Integration", OOPSLA 1994.
- J. Eifrig, S. Smith, V. Trifonov, A. Zwarico, "An Interpretation of Typed OOP in a Language with State", Lisp and Symbolic Computation, 8 (4), 1995.
- J. Eifrig, S. Smith, V. Trifonov, A. Zwarico, "A Simple Interpretation of OOP in a Language with State", Workshop on State in Programming Languages, 1993.

3 Supported Students

Elizabeth Leonard, Jonathan Eifrig, Valery Trifonov